Automata

Idle

Unload

Finished

Load   Unload

Finished

Start

Loaded   Processing

Stop

$$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$$

- Described by events and states

- Graphic

- Known algorithms

- Enormous state spaces

Automata − States
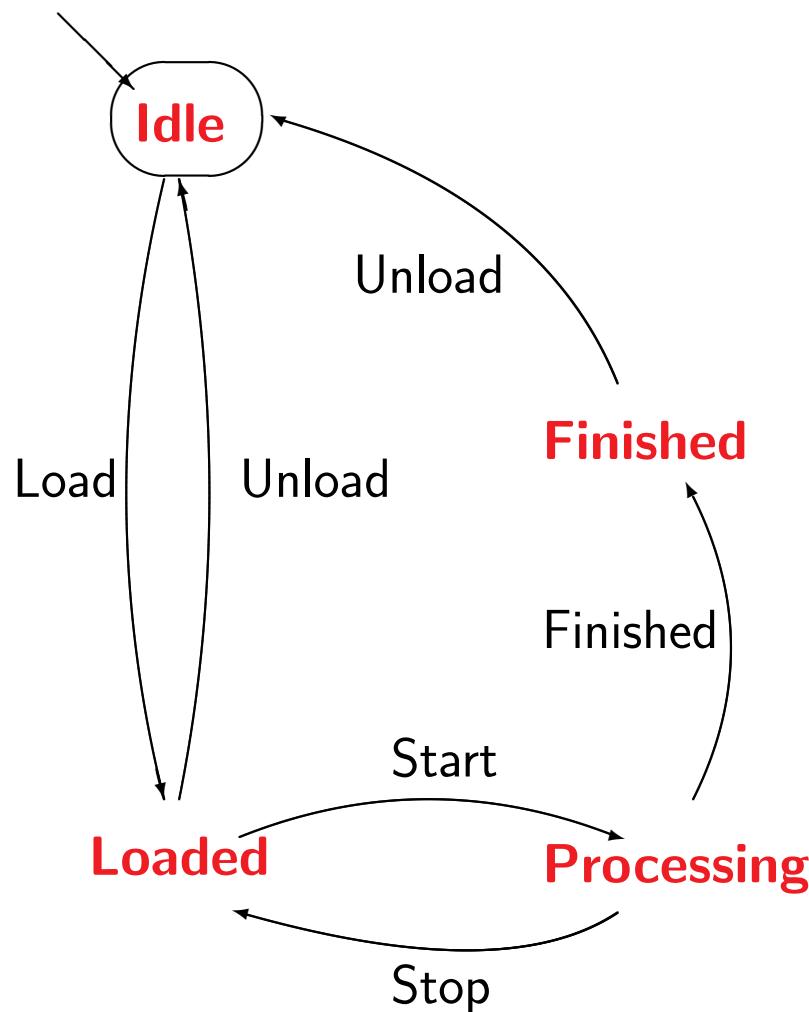
- States Q_A

  - Significant phase
  - Typically determines the output
  - State changes are transitions

- Classification

  - Initial
  - Marked
  - Unmarked
  - Forbidden

$$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$$
$$Q_A := \{\textbf{Idle}, \textbf{Loaded}, \textbf{Processing}, \textbf{Finished}\}$$

Automata − Events
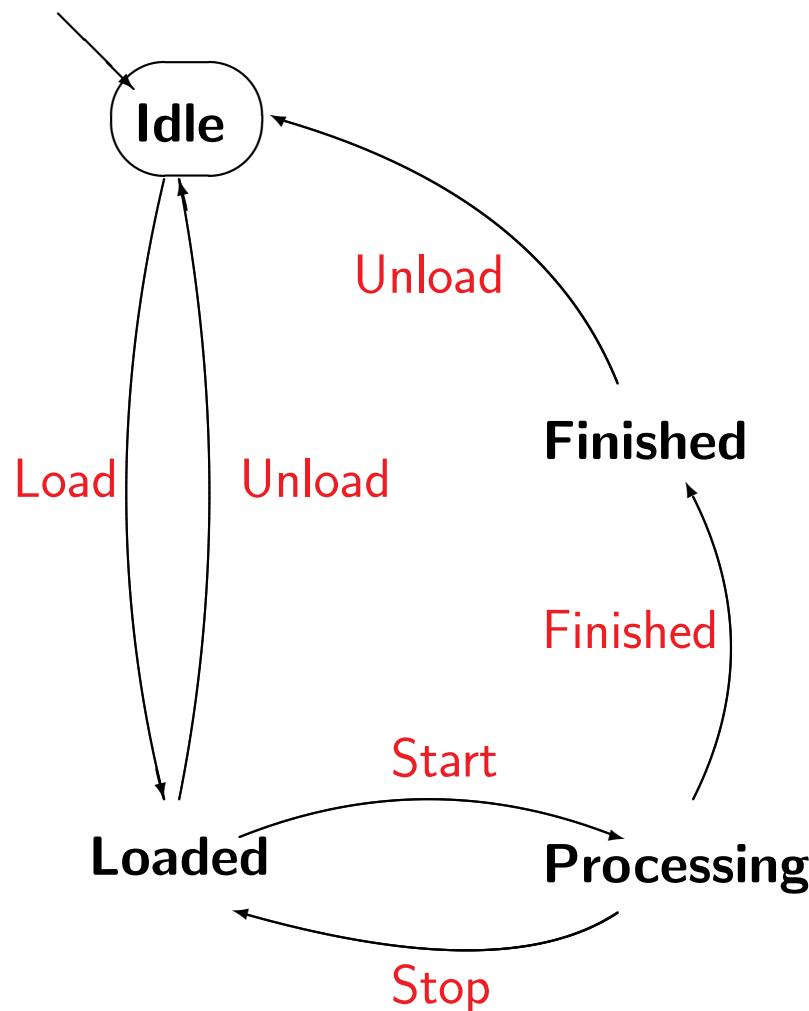
$$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$$
$$\Sigma_A := \{\text{Load}, \text{Unload}, \text{Start}, \text{Stop}, \text{Finished}\}$$

- Events Σ_A

  - Significant occurrence
  - Are often (but not always) triggered by inputs
  - Associated with transitions

- Alphabet

  - The set of possible events
  - Each automaton has its own alphabet (but events can be shared)

- Classification

  - Controllable - Uncontrollable
  - Observable - Unobservable

Automata – Transitions
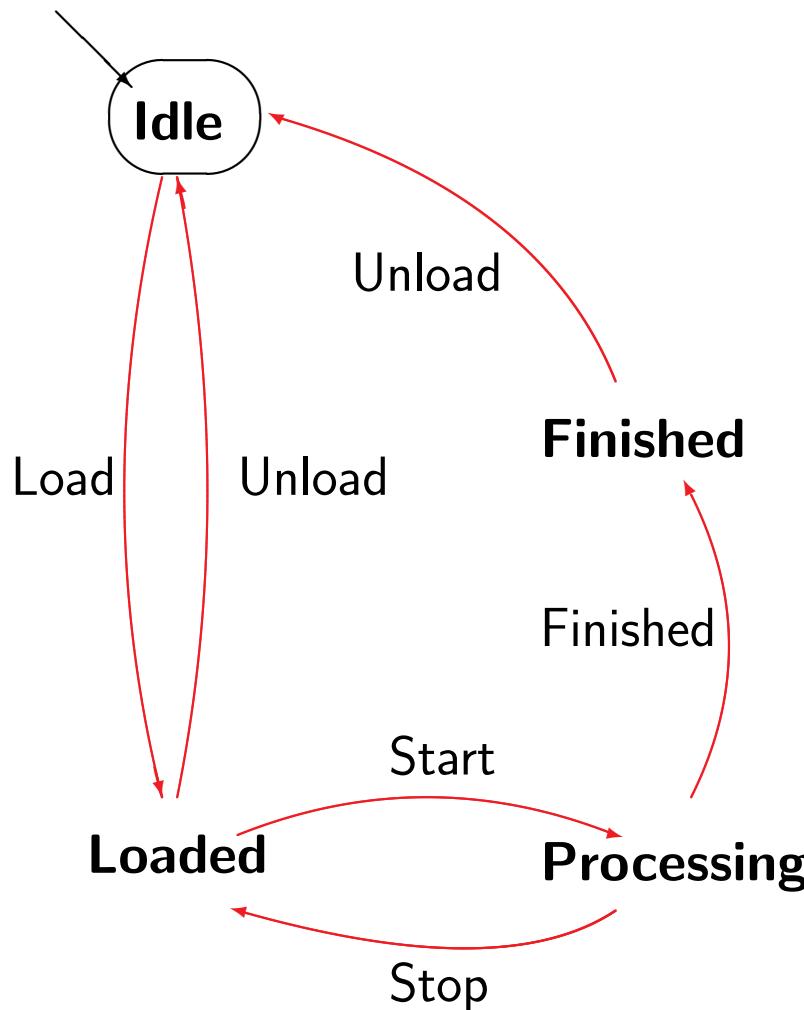
- Transitions δ_A

  - State change
  - Associated with events

- Partial function

  - Not defined for all combinations of state and event

$$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$$
$$\delta_A : Q_A \times \Sigma_A \to Q_A$$

Automata – Initial state
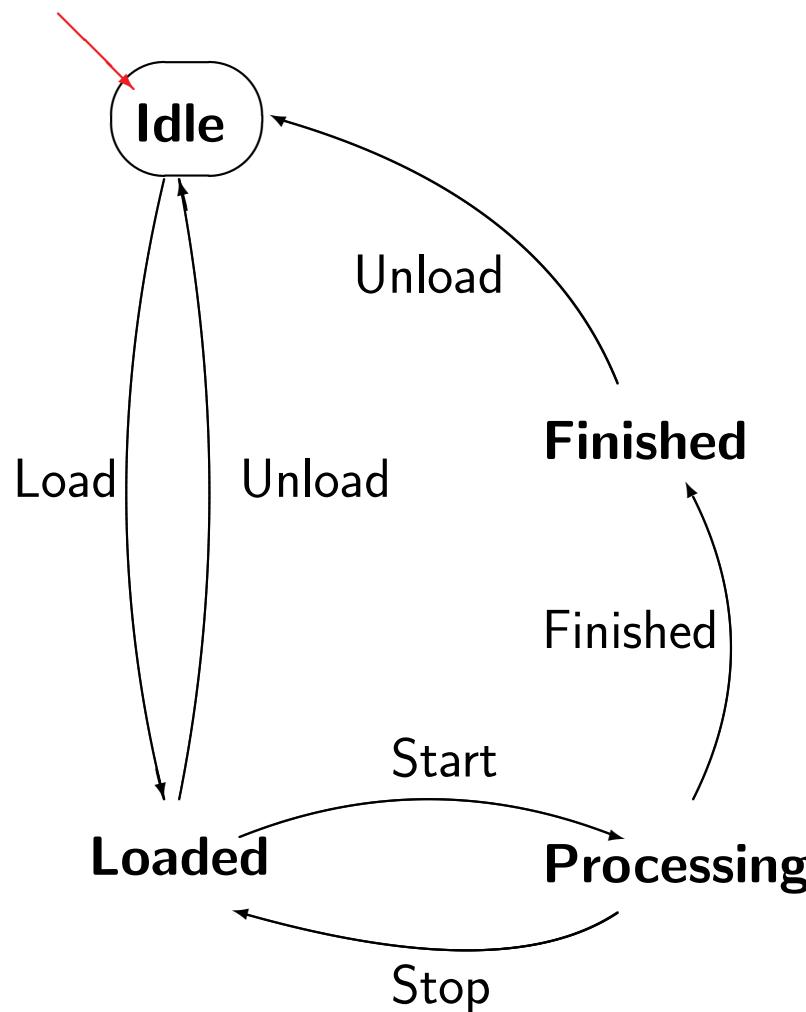
- Initial state i_A

  - Where the automaton starts

$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$

$i_A := \{\text{Idle}\}$

Automata – Marked state
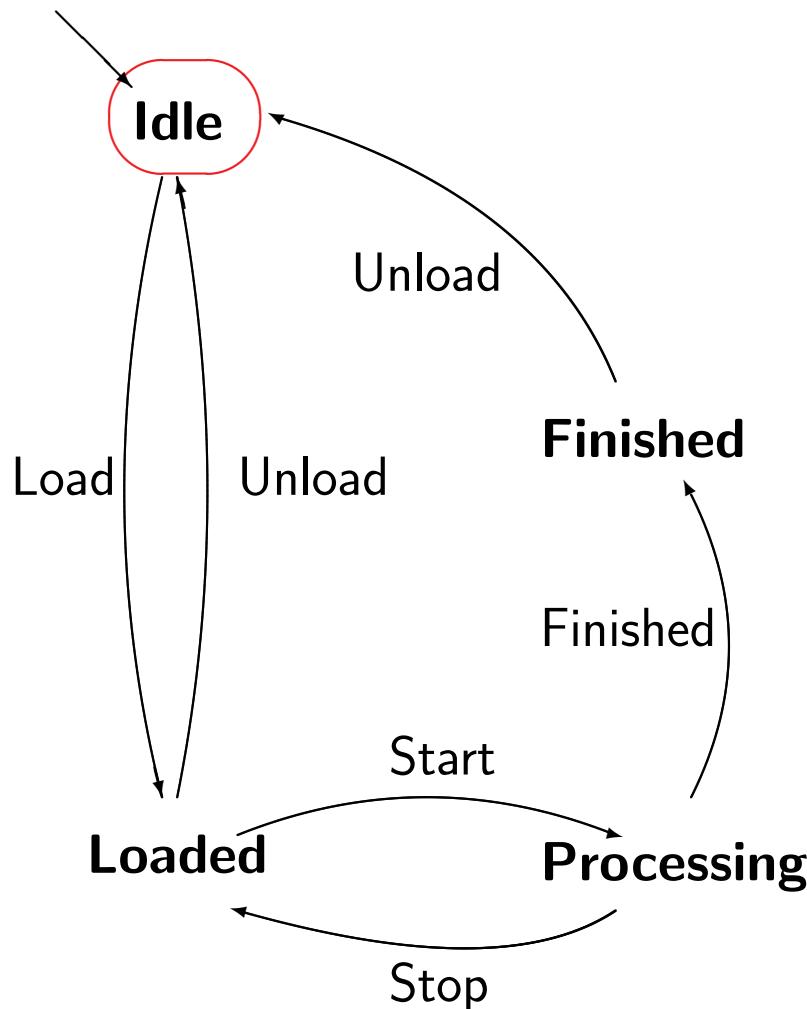
- Marked state M_A

  - Significant sub-goal
  - Can represent "cycle finished"
  - Important part of a specification
  - Means "allowed end-state"

- Typical examples

  - Initial state
  - End state
  - All states

$$A := \langle Q_A, \Sigma_A, \delta_A, i_A, M_A \rangle$$
$$M_A := \{\text{Idle}\}$$

# Example 3 – Alarm system (from Logic Control)

Alarm signal u_1 ⟶ ⟶ y_1 Siren

Control system

Acknowledgement u_2 ⟶ ⟶ y_2 Alarm light

Events $= u_1 u_2$

States $= u_1 u_2 / y_1 y_2$
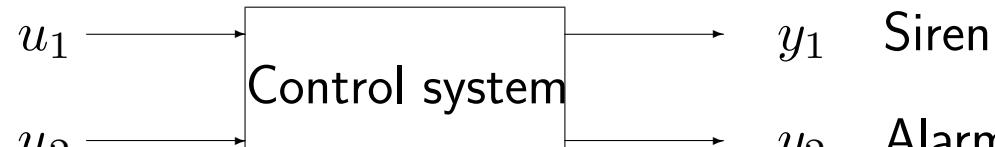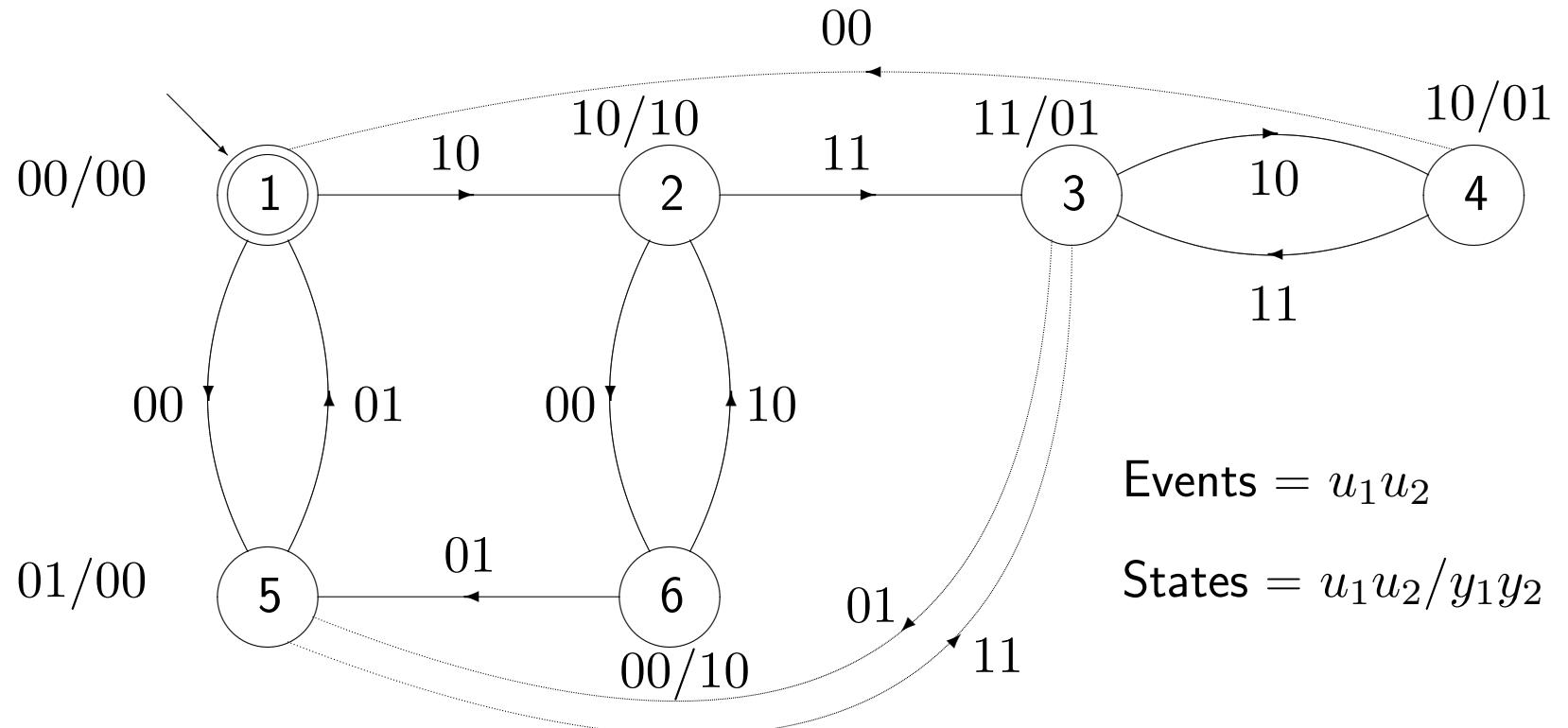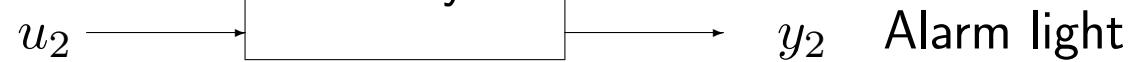
Example – PIN code reader

Example 4. A PIN (Personal Identification Number) code is a (usually) 4 digit number that is meant to identify a person or a group of persons. For example, entrances to buildings can be unlocked by punching a certain PIN code on a numeric terminal. In that case, the door unlocks if the specific code is entered, irrespective of the numbers punched before or after the sequence. Model a PIN-code reader that accepts the pin 5521 with an automaton.

Example – man, wolf, goat, cabbage

Example 5. A man together with a wolf, a goat and a cabbage head is on the left bank of a river. There is a boat large enough to carry the man and only one of the other three. The man and his company want to cross the river to the right shore. But the man can neither leave the wolf and the goat alone, nor the goat and the cabbage, because that would result in one eating the other.

Model the "system" with a set of automata, and design a supervisory controller that gets all four to the other shore without anyone being eaten.

Blocking – deadlock and livelock

Blocking is something we want to avoid, and we will start off with a verbal description:

An automaton is said to be **blocking** if *deadlock* or *livelock* can happen.

We have to define what we mean with deadlock and livelock:

Deadlock in an automaton means that there is an unmarked state where no events are possible.

The automaton jams in a state that we haven't specified as an possible end-state.

Livelock in an automaton means that there is set of unmarked states with no transition out of this set

The automaton jams in an infinite loop.

Example of blocking
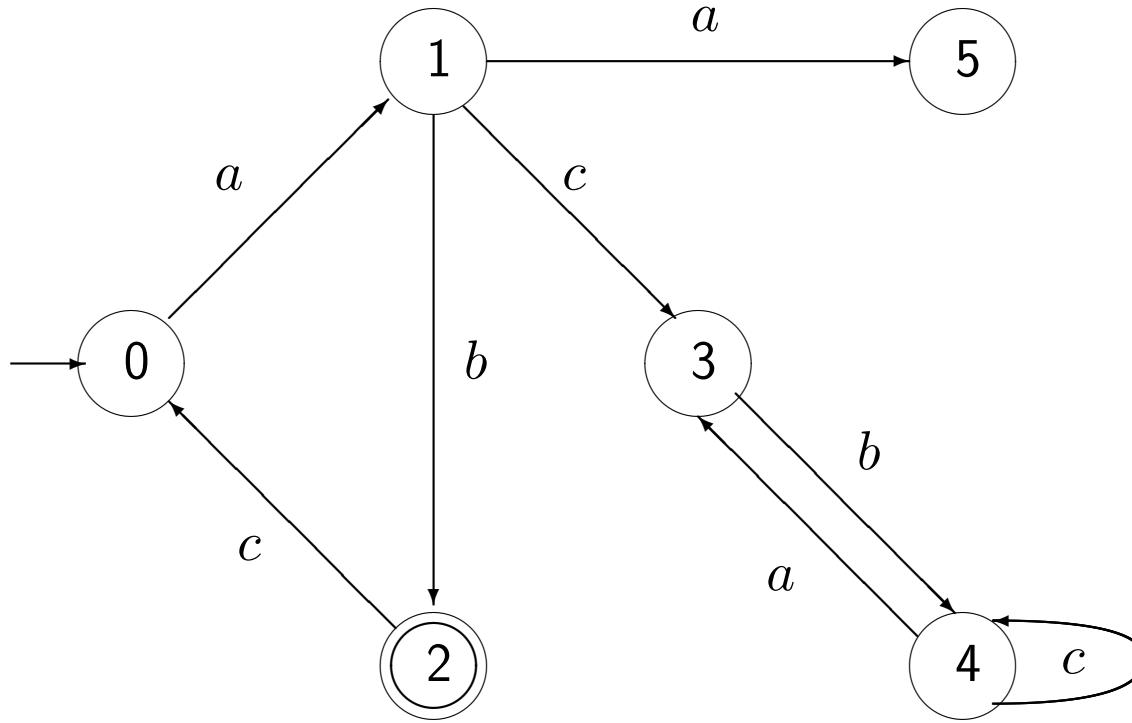
You have both deadlock and livelock in the above automaton, can you find them?

Examples of deadlock

Example 6. Dining Philosophers. Two philosophers sit around a table, either eating or thinking. The philosophers need two sticks for eating, and there are only two sticks on the table. If both philosophers pick up one stick, they both start waiting for the other to put down the other stick. You have a deadlock.

Example 7. Traffic jam. In a four-way intersection with equal priorities is the right-hand rule applied in Finland. This means that when two cars arrive about the same time the car coming from the right is allowed to drive through the intersection first. If cars arrive from all directions in about the same time nobody is allowed to drive through, we have a deadlock.

Livelock in telephony

Example 8. Assume three subscribers, A, B, and C, do the following call forwarding:

1. A forwards his calls to B.
2. B forwards his calls to C.
3. C forwards his calls to A

If you call A, B or C you would end up in an endless loop of forwarding; you have livelock.